UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/692,361	10/22/2003	Peter-Pike J. Sloan	3382-66857	9370

26119 7590 07/28/2009

KLARQUIST SPARKMAN LLP
121 S.W. SALMON STREET
SUITE 1600
PORTLAND, OR 97204

EXAMINER	
BROOME, SAID A	

ART UNIT	PAPER NUMBER
2628	

MAIL DATE	DELIVERY MODE
07/28/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

**BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES**

Application Number: 10/692,361
Filing Date: October 22, 2003
Appellant(s): SLOAN ET AL.

Stephen Wight

For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 4/30/09 appealing from the Office action mailed 11/26/2007.

(1) Real Party in Interest

The Real Party in Interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

6,333,742	MORIOKA et al	12-2001
6,628,298	DEBEVEC	09-2003
6,650,327	AIREY et al	11-2003
6,791,544	HONG et al	09-2004
2003/0063096	BURKE	04-2003
2005/0041023	GREEN	02-2005

Arvo, J., Dutre, P., Keller, A., Jensen, H., Owen, A., Phar, M., Shirley, P., Monte Carlo Ray Tracing, July 2003, SIGGRAPH 2003, pp. 1-5, 13-42.

Foley et al., Computer Graphics: Principles and Practice, 1990, Addison-Wesley, Second Edition, pp. 723-725, 741-745, 751-753, 761-763.

Purcell, T., Buck, I., Mark, W., Hanrahan, P., Ray Tracing on Programmable Graphics Hardware, July 2002, ACM Transactions on Graphics, Proceedings of ACM SIGGRAPH 2002, pp. 703-712.

Sloan, P., Kautz, J., Snyder, J., “Precomputed Radiance Transfer for Real-Time Rendering in Dynamic, Low-Frequency Lighting Environments”, SIGGRAPH 2002, pp. 527-536.

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in sec. 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1, 3, 4, 6-8, 10-12 and 20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Sloan et al.(hereinafter "Sloan", "*Precomputed Radiance Transfer for Real-Time Rendering in Dynamic, Low-Frequency Lighting Environments*") in view of Burke (US 2003/0063096) in further view of Purcell et al.(hereinafter "Purcell", "*Ray Tracing on Programmable Graphics Hardware*").

Regarding claim 1, Sloan teaches a method of producing radiance transfer coefficients for a set of points over a modeled object for rendering images of the object on a computer (sec. 6 1st ¶ lines 1-3) having a graphics processing unit for performing operations over sets of data values contained in textures (sec. 10 2nd ¶ lines 12-15), comprising iteratively for a set of points for each of a set of directions sampled about the object computing radiance transfer functions (pg. 1 2nd col. 2nd ¶ lines 1-8: "*…object's…can be viewed as a transfer function, mapping incoming and outgoing radiance…along each direction.*" and in sec. 10 1st ¶ lines : "*…the…model provides global illumination effects…in real-time…sampled every frame and at multiple points…*") and rendering the object from the direction to produce a shadow buffer representing depth from the

object in the direction for the set of points (in the caption of Fig. 2 lines 4-6: "…a *particular point on the surface represents how the surface responds to…light at that point, including…self-shadowing*…" and on pg. 2 1st col. 3rd ¶ lines 1-3: "*Shadow maps, containing depths from the light source's point of view*…"). Sloan also teaches using a graphics processing unit for determining radiance transfer contribution of the set of sampled points for the currently iterated direction based on the determined cosine terms and shadowing (pg. 1 2nd col. 2nd ¶ lines 2-8 – 3rd ¶ lines 5-7: "…*object's shaded "response" to its environment can be viewed as a transfer function, mapping incoming to outgoing radiance, which…performs a cosine-weighted integral…captures how a…object shadows itself…along each direction…graphics hardware can…sample incident radiance at a number of points.*"), where the radiance transfer is computed for all directions in response to an integral comprising the cosine terms and the effect of shadowing, and also teaches accumulating the radiance transfer contributions of the set of sampled points for the currently iterated direction with that of previously iterated direction (pg. 5 2nd col. 2nd ¶ lines 1-7: "*For diffuse surfaces, at each point…we…compute the transfer vector by SH-projecting…over the direction samples s_d, summing into an accumulated transfer*…"), in which for each point on the surface, the radiance transfer is accumulated for all directions, thereby computing the radiance transfer for the current direction as well as any previous direction. Sloan also teaches producing a radiance transfer value for each of the sampled points from the accumulated radiance transfer contributions for the iterated directions at the respective sample points (sec. 5 2nd col. 2nd ¶ lines 2-7 – 3rd ¶ lines 1-3: "…*at each point p…we further compute the transfer vector by SH-projecting Mp…at each point p…sums over all sd at every p.*"), in which a radiance transfer value is accumulated for each direction at every point, and also

illustrates a rendered image of an object in a lighting environment based on accumulated radiance transfer contribution to present the image (right portion of Fig. 1). However, Sloan fails to teach creating an object positions and normal texture for the set of sampled points mapped into the texture space. Burke teaches creating an object positions texture representing positions of a set of points sample over the object mapped into a texture space and object normals texture representing normals of the set of sampled points mapped into the texture space containing a set of data values (¶ 0035 lines 4-10: "*For each sampled point, the model data receiving unit 110 receives data representing the position of the point (e.g., "XYZ")...and normal information (e.g., "IJK")*"). However, though Sloan teaches iteratively calculating the radiance for each point for a plurality of directions (Fig. 3 of applicant's specification), in contrast of for each direction on the surface calculating the radiance for each point on the surface (Fig. 4 of applicant's specification), it would have been obvious to one of ordinary skill in the art at the time of invention to combine the teachings of Sloan and Burke with Purcell to enable reversal of the outer and inner loops of the radiance transfer method (Fig. 3 of the applicant's Specification), because reversal of the loops would optimize the radiance processing when executed on graphics hardware (Purcell: sec. 3.2 4th ¶ lines 1-5 – 10-14: "*...we present an optimization to minimize the total number of passes...There are various strategies for nesting these loops...For graphics hardware...The following is a more efficient algorithm...*"), where the nested loops are reversed to enable efficient execution on a graphics processing unit (pseudo-code in sec. 3.2 in the right col.), as disclosed in the claimed method (Fig. 4 & on pg. 8 lines 26-28 of applicant's specification). Therefore it would have been obvious to one of ordinary skill in the art at the time of invention to combine the teachings of Sloan, Burke and Purcell because this combination would provide a

radiance transfer process that iterates over a set of directions in the outer loop and iterates over sample points in the inner loop, thereby efficiently computing realistic self-shadowing and lighting effects through optimization of the process for improved functionality on graphics hardware.

Regarding claim 3, Sloan teaches graphics hardware processing software that computes radiance transfer (pg. 1 2nd col. 3rd ¶ lines 1-7, pg. 8 1st col. 4th ¶ lines 8-10 and on pg. 7 sec. 8 2nd ¶ lines 4-7 and 2nd col. lines 1-3), therefore the programming code, or software, must be embodied on a computer readable media because it is executed to render images of an object, (pg. 8 1st col. 2nd ¶ lines 1-3: "*…images in this paper (Figures 1 and 3-12) all of which were computed with the PC graphics hardware.*"). Sloan teaches programming code for performing the contents of this reference (pg. 1 1st col. 1st ¶ lines 1-9: "*…global transport simulator creates functions over the object's surface…At run-time, these transfer functions are applied…*" and on pg. 7 sec. 8 2nd ¶ lines 4-7 and 2nd col. lines 1-3: "*Our current implementation precomputes the transfer matrix p M at each point…we perform the matrix transform from equation (9) in software at each point…The result is a volume texture containing coefficients of transferred radiance…*"), therefore code or software is utilized to perform all the succeeding limitations. Sloan also teaches rendering the object from the direction to produce a shadow buffer representing depth from the object in the direction for the set of points (pg. 2 1st col. 3rd ¶ lines 1-3: "*Shadow maps, containing depths from the light source's point of view…*" and in the caption of Fig. 2 lines 4-6: "*…a particular point on the surface represents how the surface responds to incident light at that point, including global transport effects like self-shadowing…*"). Sloan also teaches determining radiance transfer contribution of the set of sampled points for the currently

iterated direction based on the determined cosine terms and shadowing on pg. 1 2nd col. 2nd ¶ lines 2-8: "*…object's shaded "response" to its environment can be viewed as a transfer function, mapping incoming to outgoing radiance, which in this case simply performs a cosine-weighted integral. A more complex integral captures how a concave object shadows itself, where the integrand is multiplied by an additional transport factor representing visibility along each direction.*"), where the radiance transfer is computed for all directions in response to a integral comprising the cosine terms and shadowing effects. Sloan teaches accumulating the radiance transfer contributions of the set of sampled points for the currently iterated direction with that of previously iterated direction (pg. 5 2nd col. 2nd ¶ lines 1-7: "*For diffuse surfaces, at each point pεO we further compute the transfer vector by SH-projecting…SH-projection to compute the transfers is performed by numerical integration over the direction samples s_d, summing into an accumulated transfer…*"), in which for each point on the surface the radiance transfer is accumulated for all directions and therefore computes the radiance transfer for the current as well as any previous direction. Sloan illustrates a rendered image of an object in a lighting environment based on accumulated radiance transfer contribution (right portion of Fig. 1). However, Sloan fails to teach creating an object positions and normal texture. Burke teaches creating an object positions texture representing positions of a set of points sample over the object mapped into a texture space and object normals texture representing normals of the set of sampled points mapped into the texture space (¶ 0035 lines 4-10: "*For each sampled point, the model data receiving unit 110 receives data representing the position of the point (e.g., "XYZ") the color of the point (including vertex color, mapped textures, and static lighting effects) (e.g., "RGB"), and normal information (e.g., "IJK")*"). However, though Sloan in view of Burke

teaches iteratively calculating the radiance in a plurality of directions for each point, it would have been obvious to one of ordinary skill in the art at the time of invention to reverse the outer and inner loops of the illustrated method (Fig. 3 of the applicant's Specification), because reversal of the loops would optimize the radiance processing when executed on graphics hardware (Purcell in sec. 3.2 4th ¶ lines 1-5 – 10-14: "*…we present an optimization to minimize the total number of passes…There are various strategies for nesting these loops…The following is a more efficient algorithm…*"), where the nested loops are reversed to enable efficient execution on a graphics processor (pseudo-code in sec. 3.2 in the right col.). Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to combine the teachings of Sloan, Burke and Purcell because this combination would provide a radiance transfer process that iterates over a set of directions in the outer loop and iterates over sample points in the inner loop, thereby efficiently computing realistic self-shadowing and lighting effects through optimization of the process for improved functionality on graphics hardware.

Regarding claim 4, Sloan teaches determining cosine terms (pg. 1 2nd col. 2nd ¶ lines 2-8: "*…object's shaded "response" to its environment can be viewed as a transfer function, mapping incoming to outgoing radiance, which in this case simply performs a cosine-weighted integral.*"), where the radiance transfer is computed for all directions in response to a integral comprising computed cosine terms applied to an integral, determining shadowing (pg. 2 1st col. 3rd ¶ lines 1-3: "*Shadow maps, containing depths from the light source's point of view…*" and in the caption of Fig. 2 lines 4-6: "*…a particular point on the surface represents how the surface responds to incident light at that point, including global transport effects like self-shadowing…*"), and determining and accumulating radiance transfer contributions over the sampled points in each

direction (caption of Fig. 2 lines 1-10: "*A transfer vector at a particular point on the surface represents how the surface responds to incident light at that point…matrix transforms the lighting coefficients into the coefficients of a spherical function…The result is convolved with the model's BRDF kernel and evaluated at the view-dependent reflection direction to yield the result at one point…*"). However, though Sloan in view of Burke teaches calculating for each point, the radiance in a plurality of directions, it would have been obvious to one of ordinary skill in the art at the time of invention to reverse the outer and inner loops of the illustrated method (Fig. 3 of the applicant's Specification), because reversal of the loops would optimize the radiance processing when executed on graphics hardware (Purcell: sec. 3.2 4th ¶ lines 1-5 – 10-14: "…*we present an optimization to minimize the total number of passes…There are various strategies for nesting these loops…The following is a more efficient algorithm…*"), where the nested loops are reversed to enable efficient execution on a graphics processor (pseudo-code in sec. 3.2 in the right col.). The motivation to combine the teachings of Sloan, Burke and Purcell is equivalent to the motivation of claim 1.

Regarding claim 6, Sloan fails to teach the limitations. Burke teaches an object positions texture arrangement of data values representing the position of each of the sampled points mapped into the texture space (¶ 0035 lines 4-10: "*For each sampled point, the model data receiving unit 110 receives data representing the position of the point (e.g., "XYZ") the color of the point (including vertex color, mapped textures, and static lighting effects) (e.g., "RGB"), and normal information (e.g., "IJK")*"). The motivation to combine the teachings of Sloan, Burke and Purcell is equivalent to the motivation of claim 1.

Regarding claim 7, Sloan fails to teach the limitations. Burke teaches that the object positions texture is stored in an RGB component format (¶ 0035 lines 4-10: "*For each sampled point, the model data receiving unit 110 receives data representing the position of the point (e.g., "XYZ") the color of the point (including vertex color…(e.g., "RGB")*"). The motivation to combine the teachings of Sloan, Burke and Purcell is equivalent to the motivation of claim 1.

Regarding claim 8, Sloan fails to teach the limitations. Burke teaches object normals texture contains an arrangement of data values representing the surface normal at each of the sampled points mapped into the texture space (¶ 0035 lines 4-10: "*For each sampled point, the model data receiving unit 110 receives data representing the position of the point (e.g., "XYZ")…and normal information (e.g., "IJK")*"). The motivation to combine the teachings of Sloan, Burke and Purcell is equivalent to the motivation of claim 1.

Regarding claim 10, Sloan teaches determining cosine terms (pg. 1 2nd col. 2nd ¶ lines 2-8: "*…object's shaded "response" to its environment can be viewed as a transfer function, mapping incoming to outgoing radiance, which in this case simply performs a cosine-weighted integral.*"), where the radiance transfer is computed for all directions in response to a integral comprising computed cosine terms applied to an integral, determining shadowing (pg. 2 1st col. 3rd ¶ lines 1-3: "*Shadow maps, containing depths from the light source's point of view…*" and in the caption of Fig. 2 lines 4-6: "*…a particular point on the surface represents how the surface responds to incident light at that point, including global transport effects like self-shadowing…*"), and determining and accumulating radiance transfer contributions over the sampled points in each direction (caption of Fig. 2 lines 1-10: "*A transfer vector at a particular point on the surface represents how the surface responds to incident light at that point…matrix*

transforms the lighting coefficients into the coefficients of a spherical function…The result is convolved with the model's BRDF kernel and evaluated at the view-dependent reflection direction to yield the result at one point…"). Sloan also teaches performing radiance transfer contributions on a pixel shader (pg. 7 1st col. 5th ¶ lines 4-7 and 2nd col. 1st ¶ lines 1-5: "*At run-time, we perform the matrix transform from equation (9) in software at each point in the volume and upload the result to the graphics hardware. The result is a volume texture containing coefficients of transferred radiance (L'_p) which is applied to R. Then in a pixel shader this transferred radiance is used to light the receiver.*"), executable on a programmable graphics processing unit, such as the programmable graphics processor (pg. 6 1st col. sec. 6 3rd ¶ line 6: "*DirectX 8.1 pixel shaders*"). Therefore one of ordinary skill would have been capable of also determining cosine terms and shadowing using the pixel shader because they both contribute to the rendering of the radiance transfer.

Regarding claim 11, Sloan teaches rendering the object from the direction comprises the object as an orthographic camera projection whose view direction is set to the current direction (pg. 2 1st col. 1st ¶ lines 14-16: "*…evaluated at the view-dependent reflection direction to produce the final shading.*"), where the shading is perform based on the view therefore the object is rendered based on the particular view direction (pg. 6 1st col. sec. 6 1st ¶ line 3 and step 4: "*Rendering O requires the following steps at run-time:… the radiance vector resulting from step 3 is convolved with O's BRDF at p, and then evaluated at the view-dependent reflection direction R*").

Regarding claim 12, Sloan teaches that the occlusion and shadowing values of the points on the object (pg. : "*Real-time, realistic global illumination… it requires integration over the*

hemisphere of lighting directions at each point (light integration), and it must account for bouncing/occlusion effects, like shadows, due to intervening matter along light paths from sources to receivers (light transport complexity)." and on pg. 5 2nd col. 1st ¶ lines : "*We tag each direction sd with an occlusion bit, 1 ( ) p dV s - , indicating whether sd is in the hemisphere and intersects O again (i.e., is self-shadowed by O).*"), therefore the depth of the sampled point is tested to determine visibility of the current sampled point in the current direction. However, Sloan fails to teach a sampled point represented in the object positions texture. Burke teaches an object positions texture that represents the depth of the sampled points (¶ 0035 lines 4-10: "*For each sampled point, the model data receiving unit 110 receives data representing the position of the point (e.g., "XYZ") the color of the point (including vertex color, mapped textures, and static lighting effects) (e.g., "RGB"), and normal information (e.g., "IJK")*"). The motivation to combine the teachings of Sloan, Burke and Purcell is equivalent to the motivation of claim 1.

Regarding claim 20, Sloan teaches determining cosine terms (pg. 1 2nd col. 2nd ¶ lines 2-8: "*...object's shaded "response" to its environment can be viewed as a transfer function, mapping incoming to outgoing radiance, which in this case simply performs a cosine-weighted integral.*"), where the radiance transfer is computed for all directions in response to a integral comprising computed cosine terms applied to an integral, determining shadowing (pg. 2 1st col. 3rd ¶ lines 1-3: "*Shadow maps, containing depths from the light source's point of view...*" and in the caption of Fig. 2 lines 4-6: "*...a particular point on the surface represents how the surface responds to incident light at that point, including global transport effects like self-shadowing...*"), and determining and accumulating radiance transfer contributions over the sampled points in each direction (caption of Fig. 2 lines 1-10: "*A transfer vector at a particular*

point on the surface represents how the surface responds to incident light at that point...matrix transforms the lighting coefficients into the coefficients of a spherical function...The result is convolved with the model's BRDF kernel and evaluated at the view-dependent reflection direction to yield the result at one point..."). However, though Sloan teaches iteratively calculating for each point, the radiance in a plurality of directions, it would have been obvious to one of ordinary skill in the art at the time of invention to reverse the outer and inner loops of the illustrated method (Fig. 3 of the applicant's Specification), because reversal of the loops would optimize the radiance processing when executed on graphics hardware (Purcell in sec. 3.2 4th ¶ lines 1-5 – 10-14: "*...we present an optimization to minimize the total number of passes...There are various strategies for nesting these loops...The following is a more efficient algorithm...*"), where the nested loops are reversed to enable efficient execution on a graphics processor (pseudo-code in sec. 3.2 in the right col.). The motivation to combine the teachings of Sloan, Burke and Purcell is equivalent to the motivation of claim 3.

Claims 2, 5, 12, 13 and 15-19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Sloan, in view of Morioka et al.(hereinafter "Morioka", US Patent 6,333,742) in further view of Burke and in further view of Purcell.

Regarding claim 2, Sloan teaches hardware-accelerated processing, which is implied to be performed on a computer system, of a radiance transfer coefficients computation for a set of points sampled over a modeled object (pg.1 2nd col. 3rd ¶ lines 1-7: "*The resulting transfer functions are represented as a dense set of vectors or matrices over its surface...The graphics hardware can dynamically sample incident radiance at a number of points.*", on pg. 8 1st col. 4th

¶ lines 8-10: "*Using graphics hardware, incident lighting can be sampled every frame and at multiple points near the object allowing dynamic, local lighting.*" and on pg. 7 sec. 8 2nd ¶ lines 4-7 and 2nd col. lines 1-3: "*Our current implementation precomputes the transfer matrix p M at each point…we perform the matrix transform from equation (9) in software at each point…The result is a volume texture containing coefficients of transferred radiance…*"), for use in rendering images of the object (Fig. 1). Sloan also teaches calculating radiance transfer using software (pg. 6 1st col. sec. 6 2nd ¶ lines 1-3: "*Step 1 can load a precomputed environment map, evaluate analytic lighting models in software, or sample radiance using graphics hardware.*"), therefore a radiance transfer processing program is executed on a computer system. Sloan teaches computing radiance transfer functions for each set of directions sampled about the object (pg. 1 2nd col. 2nd ¶ lines 1-8: " *…we have a convex, diffuse object lit by an infinitely distant environment map. The object's "response" to its environment can be viewed as a transfer function, mapping incoming and outgoing radiance…A more complex integral captures how a concave object shadows itself, where the integrand is multiplied by an additional transport factor representing visibility along each direction.*"), and also teaches rendering the object from the direction to produce a shadow buffer representing depth from the object in the direction for the set of points (pg. 2 1st col. 3rd ¶ lines 1-3: "*Shadow maps, containing depths from the light source's point of view…*" and in the caption of Fig. 2 lines 4-6: "*…a particular point on the surface represents how the surface responds to incident light at that point, including global transport effects like self-shadowing…*"). Sloan teaches determining radiance transfer contribution of the set of sampled points for the currently iterated direction based on the determined cosine terms and shadowing (pg. 1 2nd col. 2nd ¶ lines 2-8: "*…object's shaded*

"response" to its environment can be viewed as a transfer function, mapping incoming to outgoing radiance…along each direction."), where the radiance transfer is computed for all directions in response to a integral comprising the cosine terms and shadowing effects, and teaches accumulating the radiance transfer contributions of the set of sampled points for the currently iterated direction with that that of previously iterated direction (pg. 5 2nd col. 2nd ¶ lines 1-7: "*For diffuse surfaces, at each point pεO we further compute the transfer vector by SH-projecting …SH-projection to compute the transfers is performed by numerical integration over the direction samples* s_d, *summing into an accumulated transfer…*"), in which for each point on the surface the radiance transfer is accumulated for all directions and therefore computes the radiance transfer for the current as well as any previous direction, and illustrates a rendered image of an object in a lighting environment based on accumulated radiance transfer contribution (right portion of Fig. 1). However, Sloan fails to teach a memory for storing program code of at least on pixel shader, a central processing unit to execute the radiance transfer coefficients processing program and a graphics processing unit programmable by and operating to execute the at least one pixel shader and an object positions and normals texture. Morioka teaches a memory for storing program code of at least one pixel shader and a radiance transfer coefficients processing program (col.17 lines 39-47) where program code, (Fig. 21), which is known in the art to be stored on a computer readable medium, performs pixel shading in step 7 and processes radiance in step 1 where the light intensity values are determined. Morioka also teaches a central processing unit operating to execute the radiance transfer coefficients processing program (col.16 lines 10-16) where the light source information, which includes the radiance value of the pixel, is process by the CPU (Fig. 16: element 1). Morioka teaches a graphics processing unit (col.7 lines

24-33) where a geometry processor 2 (Fig. 6), performs graphics processing and executes at least one pixel shader by a rendering processor (col.7 lines 34-38, Fig. 6: element 32), and teaches the at least one pixel shader executing on a graphics processing unit performing texture operation from each direction sample about the object (col. 12 lines 42-51) where the texture generator, which comprises the rendering processor that performs the same functionality of the disclosed graphics processing unit, performs texture operations for each pixel. The generated data is then sent to the shading circuit, which performs shading, and light intensity operations for each of a set of direction about the object (col. 9 lines 1-4). However, Sloan and Morioka fail to teach an object positions and normals texture that contains data values. Burke teaches creating an object positions texture representing positions of a set of points sample over the object mapped into a texture space and object normals texture representing normals of the set of sampled points mapped into the texture space containing data values (¶ 0035 lines 4-10: "*For each sampled point, the model data receiving unit 110 receives data representing the position of the point (e.g., "XYZ")...mapped textures...and normal information (e.g., "IJK")*"). However, though Sloan teaches iteratively calculating for each point, the radiance in a plurality of directions, it would have been obvious to one of ordinary skill in the art at the time of invention to reverse the outer and inner loops of the illustrated method (Fig. 3 of the applicant's Specification), because reversal of the loops would optimize the radiance processing when executed on graphics hardware (Purcell in sec. 3.2 4th ¶ lines 1-5 – 10-14: "*...we present an optimization to minimize the total number of passes...There are various strategies for nesting these loops...For graphics hardware...The following is a more efficient algorithm...*"), where the nested loops are reversed to enable efficient execution on a graphics processor (pseudo-code in sec. 3.2 in the right col.).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to combine the teachings of Sloan, Morioka, Burke and Purcell because this combination would provide a radiance transfer process that iterates over a set of directions in the outer loop and iterates over sample points in the inner loop, thereby efficiently computing realistic self-shadowing and lighting effects through optimization of the process for improved functionality on graphics hardware.

Regarding claim 5, Sloan teaches determining cosine terms (pg. 1 2nd col. 2nd ¶ lines 2-8: "*…object's shaded "response" to its environment can be viewed as a transfer function, mapping incoming to outgoing radiance, which in this case simply performs a cosine-weighted integral.*"), where the radiance transfer is computed for all directions in response to a integral comprising computed cosine terms applied to an integral, determining shadowing (pg. 2 1st col. 3rd ¶ lines 1-3: "*Shadow maps, containing depths from the light source's point of view…*" and in the caption of Fig. 2 lines 4-6: "*…a particular point on the surface represents how the surface responds to incident light at that point, including global transport effects like self-shadowing…*"), and determining and accumulating radiance transfer contributions over the sampled points in each direction (caption of Fig. 2 lines 1-10: "*A transfer vector at a particular point on the surface represents how the surface responds to incident light at that point...matrix transforms the lighting coefficients into the coefficients of a spherical function…The result is convolved with the model's BRDF kernel and evaluated at the view-dependent reflection direction to yield the result at one point…*"). Therefore, as stated in the Specification (pg. 8 lines 24-28: "*As compared to the previous PRT preprocess pseudo-code 300 of Figure 3, the order of the inner and outer loops of the hardware-accelerated PRT preprocess 400 are reversed to be more suitable for GPU*

execution.”), the illustrated code (Fig. 4) is a reversed representation of the code (Fig. 3), in which when it is reversed, produces the same results of the prior art (pg. 12 lines 2-4). However, though Sloan teaches iteratively calculating for each point, the radiance in a plurality of directions, it would have been obvious to one of ordinary skill in the art at the time of invention to reverse the outer and inner loops of the illustrated method (Fig. 3 of the applicant’s Specification), because reversal of the loops would optimize the radiance processing when executed on graphics hardware (Purcell in sec. 3.2 4th ¶ lines 1-5 – 10-14: “…*we present an optimization to minimize the total number of passes…There are various strategies for nesting these loops…The following is a more efficient algorithm*…“), where the nested loops are reversed to enable efficient execution on a graphics processor (pseudo-code in sec. 3.2 in the right col.). The motivation to combine the teachings of Sloan, Morioka, Burke, and Purcell is equivalent to the motivation of claim 2.

Regarding claim 13, Sloan fails to teach the limitations. Burke teaches that the object positions texture contains an arrangement of data values representing the position of each of the sampled points mapped into the texture space (¶ 0035 lines 4-10: “*For each sampled point, the model data receiving unit 110 receives data representing the position of the point (e.g., "XYZ") the color of the point (including vertex color, mapped textures, and static lighting effects) (e.g., "RGB"), and normal information (e.g., "IJK")*“). The motivation to combine the teachings of Sloan, Morioka, Burke, and Purcell is equivalent to the motivation of claim 2.

Regarding claim 15, Sloan fails to teach the limitations. Burke teaches object normals texture contains an arrangement of data values representing the surface normal at each of the sampled points mapped into the texture space (¶ 0035 lines 4-10: “*For each sampled point, the*

model data receiving unit 110 receives data representing the position of the point (e.g., "XYZ")...and normal information (e.g., "IJK")"). The motivation to combine the teachings of Sloan, Morioka, Burke, and Purcell is equivalent to the motivation of claim 2.

Regarding claim 16, Sloan teaches that the set of directions are to uniformly distributed points on a unit sphere (pg. 2 1st col. 1st ¶ lines 12-16: "*…the coefficients of a spherical function representing self-scattered incident radiance at each point. This function is convolved with the object's BRDF and then evaluated at the view-dependent reflection direction…*" and on pg. 4 2nd col. 7th ¶ lines 1-3: "*…transfer the incident radiance Lp(s) into a whole sphere of transferred radiance…*"), where the set of points are distributed uniformly on the surface of the object, therefore the set of directions would also be computed uniformly on a unit sphere (Fig. 2), because the set of directions are computed for all the points on the object (pg. 5 2nd col. 3rd ¶ lines 1-3: "*The vector Mp or matrix p M at each point p is initialized to 0 before the shadow pass, which then sums over all s_d at every p.*", Fig. located in the 2nd col. of pg. 5).

Regarding claim 17, Sloan teaches rendering the object from the direction comprises the object as an orthographic camera projection whose view direction is set to the current direction (pg. 2 1st col. 1st ¶ lines 14-16: "*…evaluated at the view-dependent reflection direction to produce the final shading.*"), where the shading is perform based on the view therefore the object is rendered based on the particular view direction, (pg. 6 1st col. sec. 6 1st ¶ line 3 and step 4: "*Rendering O requires the following steps at run-time:… the radiance vector resulting from step 3 is convolved with O's BRDF at p, and then evaluated at the view-dependent reflection direction R*").

Regarding claim 18, Sloan teaches that the occlusion and shadowing values of the points on the object (pg. lines : "*Real-time, realistic global illumination… it requires integration over the hemisphere of lighting directions at each point (light integration), and it must account for bouncing/occlusion effects, like shadows, due to intervening matter along light paths from sources to receivers (light transport complexity).*" and on pg. 5 2nd col. 1st ¶ lines : "*We tag each direction sd with an occlusion bit…indicating whether sd is in the hemisphere and intersects O again (i.e., is self-shadowed by O).*"), therefore the depth of the sampled point is tested to determine visibility of the current sampled point in the current direction. Sloan fails to teach a sampled point represented in the object positions texture. Burke teaches an object positions texture that represents the depth of the sampled points (¶ 0035 lines 4-10: "*For each sampled point, the model data receiving unit 110 receives data representing the position of the point (e.g., "XYZ") the color of the point (including vertex color, mapped textures, and static lighting effects) (e.g., "RGB"), and normal information (e.g., "IJK")*"). The motivation to combine the teachings of Sloan, Morioka, Burke, and Purcell is equivalent to the motivation of claim 2.

Regarding claim 19, Sloan teaches code or software executable on the graphics accelerating hardware of the computer (pg.1 2nd col. 3rd ¶ lines 1-7: "*The resulting transfer functions are represented as a dense set of vectors or matrices over its surface…The graphics hardware can dynamically sample incident radiance at a number of points.*" and on pg. 7 sec. 8 2nd ¶ lines 4-7 and 2nd col. lines 1-3: "*Our current implementation precomputes the transfer matrix p M at each point…we perform the matrix transform from equation (9) in software at each point…The result is a volume texture containing coefficients of transferred radiance…*"), to perform texture –based operations is a pixel shader (pg. 7 1st col. 5th ¶ lines 4-7 and 2nd col. 1st ¶

lines 1-5: "*At run-time, we perform the matrix transform from equation (9) in software at each point in the volume and upload the result to the graphics hardware. The result is a volume texture containing coefficients of transferred radiance (L'_p) which is applied to R. Then in a pixel shader this transferred radiance is used to light the receiver.*"), executable on a programmable graphics processing unit, such as the programmable graphics processor (pg. 6 1st col. sec. 6 3rd ¶ line 6: "*DirectX 8.1 pixel shaders*").

Claim 9 is rejected under 35 U.S.C. 103(a) as being unpatentable over Sloan in view of Burke in further view of Purcell, and in further view of Arvo et al.(hereinafter "Avro", "Monte Carlo Ray Tracing").

Regarding claim 9, Sloan, Burke and Purcell fail to teach the limitations. Arvo teaches a set of directions that are generated as uniformly distributed points on a unit sphere based on a mapping from the unit square to the sphere (pg. 41 1st ¶ lines 1-5: "*…uniformly distributed samples in the unit square are mapped to uniformly distributed samples in the range. Such mappings also preserve stratification, also known as jitter sampling…*" and on pg. 23 5th ¶ lines 1-4: "*…a set of jittered points on the unit square can be easily transformed to a set of jittered points on the hemisphere…*"), where the points form the unit square are mapped on to the hemisphere or unit sphere (pg. 23 2nd ¶ lines 1-3: "*To choose reflected ray directions for zonal calculations or distributed ray tracing, we can think of the problem as choosing points on the unit sphere or hemisphere…*"). It would have been obvious to one of ordinary skill in the art at the time of invention to combine the teachings of Sloan, Burke, Purcell and Arvo because this combination would provide a smooth representation of the lighting of a surface through the use of jittered sampling, which prevents unwanted aliasing effects.

Claim 14 is rejected under 35 U.S.C. 103(a) as being unpatentable over Sloan, in view of Morioka, in further view of Burke, in further view of Purcell, and in further view of Airey et al. (hereinafter "Airey", US Patent 6,650,327).

Regarding claim 14, Sloan fails to tech the limitations. Burke teaches an object positions texture (¶ 0035 lines 4-10: "*For each sampled point, the model data receiving unit 110 receives data representing the position of the point (e.g., "XYZ") the color of the point (including vertex color, mapped textures, and static lighting effects)…*"). However, Burke, Purcell and Morioka fail to teach storing texture in a floating point number format. Airey teaches storing texture in floating point number format (col.4 lines 18-20: "*Texturing, fog, and antialiasing all operate on floating point numbers. The texture map stores floating point texel values.*"). It would have been obvious to one of ordinary skill in the art at the time of invention to combine the teachings of Sloan, Morioka, Burke, Purcell and Airey because this combination would provide an efficient storage of textures in floating point number format that enables a more accurate processing to take place on the graphics hardware.

(10) Response to Argument

The appellant argues that Purcell fails to suggest the proposed modification of an outer loops iterating over directions and inner loop iterating over points. However, the appellant's arguments are unpersuasive because Sloan was relied upon to teach iterating over several directions for a plurality of points, in which Purcell was used to modify the teachings of Sloan to teach nesting loops thereby reversing the outer and inner loops. Therefore in response to applicant's arguments against the references individually, one cannot show nonobviousness by

attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.,* 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

The appellant's argues that the proposed modification would render the prior Sloan method inoperable for its intended purpose. However, the appellant's arguments are unpersuasive because one skilled in the art would have recognized modifying the radiance transfer coefficient computation of Sloan to realize faster graphics processing through nesting the loops of the radiance transfer coefficient program for implementation on graphics hardware, as taught by Purcell. Therefore modification of Sloan would not have made the method of Sloan inoperable for its intended purpose, because it must be recognized that any judgment on obviousness is in a sense necessarily a reconstruction based upon hindsight reasoning. But so long as it takes into account only knowledge which was within the level of ordinary skill at the time the claimed invention was made, and does not include knowledge gleaned only from the applicant's disclosure, such a reconstruction is proper. See *In re McLaughlin*, 443 F.2d 1392, 170 USPQ 209 (CCPA 1971).

The appellant's argues that the Examiner fails to articulate adequate rationale how Purcell would have led the person of ordinary skill in the art to make the proposed modification. However, the appellant's arguments are unpersuasive because it would have been obvious to one of ordinary skill in the art at the time of invention to combine the radiance transfer computation of Sloan and point data of Burke with the nested loop process taught by Purcell because nested reversal of the loops would optimize the speed and efficiency of radiance processing when executed on graphics hardware, thereby efficiently computing realistic self-

shadowing and lighting effects produced from computation of the radiance transfer contribution, ensuring optimization of the radiance processing for improved functionality on graphics hardware.

The appellant's argues that the asserted art lacks texture operations on textures containing data values representing normals and positions of sample points mapped into a texture space. However, the appellant's arguments are unpersuasive because Burke clearly teaches implementing operations for textures that contain data values representing normals, as well as positions, of sample points for mapping the texturing information on a surface using a surface map to represent the mapped textures, as disclosed in ¶0009 lines 8-11 and ¶0035 lines 4-8.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Said Broome

/Said Broome/

Conferees:

Xiao Wu, Supervisory Patent Examiner, Art Unit 2628

/XIAO M. WU/

Supervisory Patent Examiner, Art Unit 2628

Kee Tung, Supervisory Patent Examiner, Art Unit 2628

/Kee M Tung/

Supervisory Patent Examiner, Art Unit 2628